Entrega 1

Jordi Vanrell, Daniel Ramos & Sergi Fornés

1) The file **facebook_sample_anon.txt** is a data table containing the list of edges of an anonymized sample of the Facebook friendship network. Download it on your computer, upload it to R as a dataframe, and define an undirected graph with this list of edges.

Leemos el archivo con los datos y lo metemos en un dataframe.

```
data <- read.table("facebook_sample_anon.txt")
head(data)
```

```
##   V1 V2
## 1  0  1
## 2  0  2
## 3  0  3
## 4  0  4
## 5  0  5
## 6  0  6
```

Creamos el grafo.

```
gf <- graph_from_data_frame(d=data, directed = F)
```

a) Is it connected? If it is not, replace it by its largest connected component.

La función `is.connected` comprueba que el grafo es conexo.

```
is.connected(gf)
```

```
## [1] TRUE
```

Observamos que es cierto.

b) Compute the edge density.

```
edge_density(gf, loops=F)
```

```
## [1] 0.01081996
```

La densidad del grafo es de 0.011. Esto significa que el 1.1% de las aristas posibles están definidas.

c) What is the mean distance among the subjects?

```r
mean_distance(gf, directed = F)
```

```
## [1] 3.692507
```

La media de la distancia más corta entre todos los pares de nodos posibles es 3.693.

d) Calculate the list of vertices in a diameter of the graph. Plot only this path with the size of the node proportional to the degree of the node.

Calculamos el diámetro.

```r
dmt <- get_diameter(gf, directed = F)
dmt
```

```
## + 9/4039 vertices, named, from 3223e8a:
## [1] 687  686  698  3437 567  414  594  3980 3981
```

Lo dibujamos. FALTA

e) Calculate the shortest path from the vertex named "1000" to the vertex named "2000" in the original file.

```r
shortest_paths(gf,
               from = V(gf)[name=="1000"],
               to  = V(gf)[name=="2000"])$vpath
```

```
## [[1]]
## + 5/4039 vertices, named, from 3223e8a:
## [1] 1000 107  58   1912 2000
```

f) Calculate a clique of 5 friends, if there is one. FALTA

```r
#sapply(cliques(gf), length)
```

h) Calculate the list of names of verteces that are the neigbours of verteces of degree one and that are not of degree one.

```r
# Ponemos el atributo degree a cada nodo
V(gf)$degree = degree(gf)

# Obtenemos los nodos con 1 vecino
alone <- V(gf)[V(gf)$degree == 1]

# Obtenemos la lista de nodos vecinos de los nodos de grado 1
vecinos <- adjacent_vertices(gf, alone)
```